Graph-of-Entity: A Model for Combined Data Representation and Retrieval

José Devezas
INESC TEC, Porto, Portugal
Faculty of Engineering, University of Porto, Portugal
http://josedevezas.com
jld@fe.up.pt

Carla Lopes
INESC TEC, Porto, Portugal
Faculty of Engineering, University of Porto, Portugal
ctl@fe.up.pt

Sérgio Nunes
INESC TEC, Porto, Portugal
Faculty of Engineering, University of Porto, Portugal
ssn@fe.up.pt

Abstract

Managing large volumes of digital documents along with the information they contain, or are associated with, can be challenging. As systems become more intelligent, it increasingly makes sense to power retrieval through all available data, where every lead makes it easier to reach relevant documents or entities. Modern search is heavily powered by structured knowledge, but users still query using keywords or, at the very best, telegraphic natural language. As search becomes increasingly dependent on the integration of text and knowledge, novel approaches for a unified representation of combined data present the opportunity to unlock new ranking strategies. We tackle entity-oriented search using graph-based approaches for representation and retrieval. In particular, we propose the graph-of-entity, a novel approach for indexing combined data, where terms, entities and their relations are jointly represented. We compare the graph-of-entity with the graph-of-word, a text-only model, verifying that, overall, it does not yet achieve a better performance, despite obtaining a higher precision. Our assessment was based on a small subset of the INEX 2009 Wikipedia Collection, created from a sample of 10 topics and respectively judged documents. The offline evaluation we do here is complementary to its counterpart from TREC 2017 OpenSearch track, where, during our participation, we had assessed graph-of-entity in an online setting, through team-draft interleaving.

2012 ACM Subject Classification Information systems → Document representation; Information systems → Retrieval models and ranking; Mathematics of computing → Graph theory

Keywords and phrases Entity-oriented search, graph-based models, collection-based graph

Digital Object Identifier 10.4230/OASIcs.SLATE.2019.1

Funding *José Devezas*: José Devezas is supported by research grant PD/BD/128160/2016, provided by the Portuguese national funding agency for science, research and technology, Fundação para a Ciência e a Tecnologia (FCT), within the scope of Operational Program Human Capital (POCH), supported by the European Social Fund and by national funds from MCTES.
Sérgio Nunes: This work is partially supported by Project "TEC4Growth – Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-000020", financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF).

8th Symposium on Languages, Applications and Technologies (SLATE 2019).
Editors: Ricardo Rodrigues, Jan Janoušek, Luís Ferreira, Luísa Coheur, Fernando Batista, and Hugo Gonçalo Oliveira; Article No. 1; pp. 1:1–1:14

OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

As the production of digital documents continues to increase, the answers we are looking for become harder to reach, particularly when relying only on identifiers and linked data to directly reach relevant content. Moreover, using a structured query language is frequently inappropriate for a regular user, who prefers natural language to express their information needs [30]. Full-text search is often the answer, but it inherently discards structure, which is extremely valuable to increase precision. In this work, we attempt to integrate unstructured text and structured knowledge in order to improve retrieval effectiveness in entity-oriented search tasks.

Search has evolved from keyword-based matching. Over time, it has grown increasingly dependent on semantic matching, largely supported on natural language understanding techniques. The need to integrate unstructured text and structured knowledge has substantially increased. In fact, one of the biggest challenges in semantic search is dealing with heterogeneity [15], in particular on the web, where a potentially unlimited number of topics exist. We tackle the problem of heterogeneity in entity-oriented search by proposing a unified graph-based model for terms and entities, where relations are seen as leads to be followed in the investigation of a given information need.

The more accurately a user's information need is identified through query understanding, and the better the information within a document is understood, the more likely the query will be matched with relevant documents or entities mentioned in those documents. This frequently results in improved retrieval effectiveness and, therefore, increased user satisfaction. What about when there is ambiguity? Can we always use entity linking to segment and semantically tag a query, discarding all other segmentations, even those which are equally likely? What if we were unable to provide an adequate answer to the user, even though the information he/she sought was available in the indexed corpus?

In the graph-of-entity, we integrate query entity linking into the ranking process, that is, a given entity in the graph is more relevant if it was reached from a nearby seed node (usually another entity) whose probability of being a good representation of the query is high (i.e., it has a high confidence weight). This probability models the certainty degree of the query entity linking process.

Listing 1 SPARQL query for the shortest path between "Axel A. Weber" and "Solingen" in DBpedia.

```
PREFIX : <http://dbpedia.org/resource/>
SELECT DISTINCT ?s ?o1 ?t
WHERE {
  VALUES ?s { :Axel_A._Weber }
  VALUES ?t { :Solingen }
  ?s [] ?o1 .
  ?o1 [] ?t
}
```

Listing 2 SPARQL query for the shortest path between "Axel Weber (athlete)" and "Solingen" in DBpedia.

```
PREFIX : <http://dbpedia.org/resource/>
SELECT DISTINCT ?s ?o1 ?o2 ?o3 ?o4 ?o5 ?t
WHERE {
  VALUES ?s { :Axel_Weber_(athlete) }
  VALUES ?t { :Solingen }
  ?s [] ?o1 .
  ?o1 [] ?o2 .
  ?o2 [] ?o3 .
  ?o3 [] ?o4 .
  ?o4 [] ?o5 .
  ?o5 [] ?t
}
```

For example, let us assume the ambiguous mention to "Axel Weber", who, according to Wikipedia, can either be the athlete or the economist. Let us now assume that the query also mentions "Solingen", which is the birthplace of "Jens Weidmann", the successor of "Axel A.

Weber", the economist. Now, the probability of "Axel Weber" referring to the economist increases, but there might also be a longer path connecting "Axel Weber", the athlete, to "Solingen". We can easily check this using DBpedia's SPARQL endpoint, by manually testing increasingly longer paths between both "Axel Weber" individuals and "Solingen". Listing 1 shows the SPARQL query for the shortest path between "Axel A. Weber" and "Solingen", which are only linked by one other entity, "Jens Weidmann" – this is consistent with what we have already described. Listing 2 shows the SPARQL query for the shortest path between "Axel Weber (athlete)" and "Solingen", which are linked by five other entities, through two distinct paths – no shorter path would link the two entities. While the query `[ axel weber solingen ]` is more likely to refer to "Axel Weber", the economist, there might still be a niche where users could be searching for "Axel Weber", the athlete, investigating whether there is a relation between the person and the location.

This type of unified approach is more prepared to take advantage of available information, discarding no lead, in order to provide the freedom to search for all matching items. We might say that word or entity disambiguation would happen "organically" during the process of ranking. The hypothesis is that this might improve effectiveness for search queries in the long tail [4], in particular by increasing recall without decreasing precision.

The remainder of this article is organized as follows. In Section 2, we do a literature review about graph-based entity-oriented search, by separately covering entity-oriented search approaches and then graph-based search approaches, closing with a discussion on common ground. In Section 3, we introduce the concept of combined data and present the INEX 2009 Wikipedia Collection. In Section 4, we introduce the technological framework we used, as well as toy example, that we use to describe our implementation of graph-of-word, an existing graph-based representation and retrieval model, as well as our own novel model for combined data, the graph-of-entity. In Section 5, we describe the evaluation approach used to compare the graph-of-word and the graph-of-entity, based on a small subset of the INEX collection. Finally, in Section 6, we present our final remarks and conclusions.

2 Graph-based entity-oriented search

About 80% of queries contain at least one entity [3] and, on average, there are 1.6 entities per sentence (based on the CoNLL 2003 English training set [26]), which makes entity-oriented search a relevant problem within information retrieval. There have been multiple approaches to entity-oriented search where graphs have been used, in particular as a way of representing knowledge bases. A well-known example is Google, who, in 2012, created Knowledge Graph [28], partially powered by Freebase [8], to improve their search engine. In the last few years, there has been work in graph-based approaches for information retrieval [7, 25], and also a growing need for unified models [13, 33, 32]. While many solutions focus on the integration of signals obtained from text represented in an inverted index with signals obtained from external knowledge bases like Wikipedia [2], there have been fewer attempts at modeling text and knowledge in an unified manner, as a single data structure. In this section, we approach the literature about graph-based entity-oriented search by covering two main aspects: entity-oriented search and graph-based search.

Entity-oriented search. Entity-oriented search is a type of semantic search that takes keyword or [telegraphic] natural language queries as input and returns the results that best match the query. Results might include metadata about a particular entity (e.g., an infobox), a list of entities represented by a small subset of relevant metadata (e.g., photo and name), a

direct answer to the user's question, and the traditional documents (e.g., web pages), usually accompanied by a summary. Notable approaches to entity-oriented search include virtual documents [3, 24], learning to rank [20, 31, 10, 9] and the integration of an inverted index and a triplestore [5].

Bautin and Skiena [3] tackled the problem of entity search based on virtual documents called concordances. Each concordance, representing an entity in an inverted index, contained all the sentences mentioning that particular entity in the corpus. The approach does not take advantage of external knowledge and it does not extract information (structured data) from the indexed corpus, both of which could be used to improve performance. The authors claim to be the *first in literature* implementation of an entity search engine. Thus, it is also natural that they used their own evaluation approach, based on the juxtaposition score, instead of available test collections. This makes it difficult to position their work in regards to the state of the art.

Chen et al. [9] presented an empirical study of several learning to rank approaches over common benchmarks, such as the Entity Search Track from SemSearch. The RankSVM [19] model is shown to be the overall best solution and, in particular for SemSearch ES, it achieves the best MAP@100, P@10, P@20 and NDCG@20, when compared with four other models (SDM-CA, MLM-CA, FSDM and Coordinate Accent [34]). Another interesting conclusion presented by the authors is that the correct identification of query types (e.g., keyword queries, long natural language question queries) is important to increase the effectiveness of learned models, in order to boost particular entity fields. Nevertheless, the goal of the graph-of-entity, which we present here, is to make such steps inherently part of the ranking process. The intuition is that the query type is implicitly represented by the structural position of the nodes that best represent the query in a graph. Similarly, instead of query entity linking, we might use the graph to take into account the probability that a given entity is linked to a part of the query, in order to better rank results.

Graph-based search. In graph-based search, particularly over text corpora, discourse properties can be captured using a graph, either per document or for the whole collection. Such graphs can represent relations between words [6, 7, 25], passages [12], or documents [23, 18], modeling similarities [12], dependencies [25], or even temporal [17] or spatial [22] dimensions.

Blanco and Lioma [6] proposed a method for term weighting based on random walks over a graph of terms, where each term was linked to other terms co-occurring in a window of size n. Similarly to PageRank [23], the weight of a term modeled the probability of jumping from that vertex to another random vertex in the undirected unweighted graph. They were able to obtain a performance comparable to TF-IDF and even outperform it for large window sizes, capturing relations between terms within a distance of 6 to 30.

Rousseau and Vazirgiannis [25] expanded on the work by Blanco and Lioma, presenting an alternative but similar approach for graph-based term weighting. They proposed a directed unweighted graph, the graph-of-word, that similarly captured relations between terms within a window of size n, but this time the target terms were required to follow the source terms instead of simply co-occurring within the window. The authors also used the indegree of the vertex instead of random walks to assign a weight to each term. Their approach significantly outperformed BM25 and, in some cases, even BM25+ [21], a lower-bounded version of BM25. Besides achieving a better performance, another reason to use their graph-based approach is that it does not require any parameter tuning or lower-bounding normalization. The graph-of-word only requires a parameter n for the window size, during indexing, which can be semantically set, since it simply captures a larger context as it grows, at the expense of efficiency.

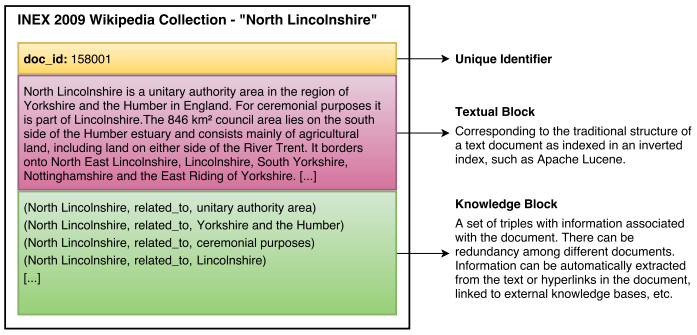


Figure 1 Extended document definition for combined data. Example from INEX 2009 Wikipedia Collection, for the XML representing the Wikipedia article about "North Lincolnshire".

Two aspects combined. From the surveyed literature, we can make two assumptions about entity-oriented search. First, structured data from knowledge bases, which are inherently representable as graphs, is a fundamental part of the semantic search process. Therefore, knowledge bases must somehow be integrated into the existing frameworks, which are mostly supported by inverted files. Many approaches exist to integrate signals from text and knowledge, but fewer common representation models have been proposed so far. Secondly, graphs have consistently been used to improve text retrieval, even outperforming weighting schemes such as BM25. Graphs can thus be used to represent text and are also frequently used to represent knowledge. It is definitely of value to study how to combine these types of graphs, in order to take advantage of the information locked within unstructured data through the integration of structured data – the knowledge base augments the text, through entities and their relations, and the text augments the knowledge base, providing leads to new information, seamlessly and through a common model (all are nodes in a graph).

What we propose is that the representation model for text and knowledge should be shared, using a graph data structure to capture discourse properties from text, relations between entities from knowledge bases, and term–entity associations based, at the very least, on potentially obvious relations between terms and entities (e.g., through substring matching). The ideal graph-based representation should: (*i*) capture information complexity, while avoiding redundancy; (*ii*) facilitate the cross-reference of information from distinct individual sources; (*iii*) propose a clear representation for combined data (text + knowledge) [2, Definition 2.3] that is easily extensible to other types of media. The open research question is whether or not such a combined data model will, through the unlocking of innovative weighting schemes, improve retrieval effectiveness. In this paper, we propose and evaluate a baseline model, the graph-of-entity, which defines a graph-based representation for combined data, as well as a graph-based weighting scheme that can be used for entity ranking. We compare the graph-of-entity with an implementation of the graph-of-word, in order to position our baseline model within the state of the art.

3 INEX 2009 Wikipedia Collection

In order to assess effectiveness, we take advantage of the INEX 2009 Wikipedia Collection [27], which includes semi-structured data from Wikipedia. The INEX 2009 Wikipedia Collection is an XML collection of Wikipedia articles, which have been annotated with over 5,800 entity classes from the YAGO [29] ontology. It contains over 2.6 million articles and requires 50.7 GB of disk space for storage, when uncompressed. The INEX Ad Hoc Track [16, 1] also

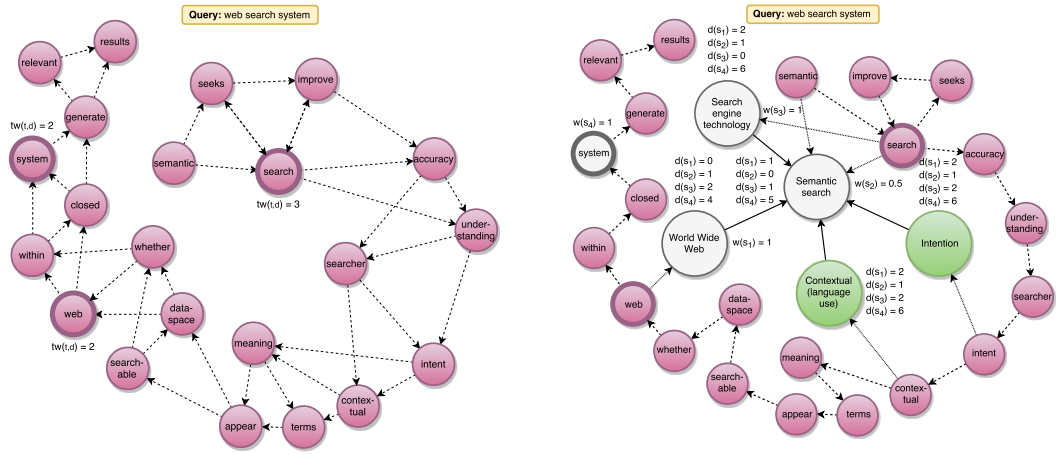
provides 115 topics from the 2009 occurrence, with 50,725 individual relevance judgments, and 107 topics from the 2010 occurrence, with 39,031 individual relevance judgments. Each individual relevance judgment contains the query identifier, the document identifier, the number of relevant characters, the offset of the best entry point (usually the first relevant passage) and offset–length pairs for the relevant passages.

As we built the graph-of-entity for all the collection, each document was represented by an entity node (i.e., the *type* attribute was set to `entity`), containing three main attributes: *doc_id*, *name* and *url*. XPath was used to extract relevant attributes. The *doc_id* was given by `//header/id/text()`, the *name* attribute was given by `//header/title/text()`, and the *url* attribute was built from the entity's Wikipedia page, based on `http://en.wikipedia.org/?curid=`<*doc_id*>. Textual content was extracted from `//bdy/descendant-or-self::*[not(ancestor-or-self::template) and not(self::caption)]/text()`. It was then tokenized, storing, for each token, a term node (i.e., the *type* attribute was set to `term`), and creating edges, with a *doc_id* attribute, between pairs of adjacent terms. For each document, links were extracted from the value given by `//link/@xlink:href` and then an entity node was created for each link, with an edge labeled `related_to`, linking the entities from the source and target documents. Further details of the representation model, including the creation of edges between term and entity nodes, will be given in Section 4.2.

Figure 1 illustrates the extracted elements from each XML document, forming what we designate as an extended document for combined data. A regular document usually contains multiple text fields (e.g., *title*, *content*, etc.), which corresponds to the textual block in the extended document. However, we also include a knowledge block, in the form of triples, that are usually available as structured data in the original document. For the INEX collection, the knowledge block can be directly extracted from the XML (we used links to other documents, in order to implicitly build the triples), but in other collections this could be obtained as the result of an information extraction pipeline. There is no restriction about the source of the knowledge block, except that it should represent a set of triples related to the document. For example, the triples might represent co-occurring entities in a sentence or paragraph, or statements obtained from a dependency parser, or they could represent external knowledge about identified entities, from an external knowledge base.

In order to evaluate retrieval over the graph-of-entity, we use the title of each topic as a search query. This is given by `//topic/title/text()` of the `2010-topics.xml` file. We then assess effectiveness based on whether or not retrieved documents contain relevant passages, according to the provided relevance judgments (`inex2010.qrels`). The evaluation process will be further detailed in Section 5.

Smaller subset: INEX 2009 10T-NL. Due to the inability of efficiently indexing the complete INEX 2009 Wikipedia Collection with our graph-based implementations, which were supported by a graph database, we were forced to lower the scale to a smaller subset of the INEX 2009 collection. Accordingly, we prepared a sampling method, based on the topics used for relevance assessment in the INEX 2010 Ad Hoc Track. In order to create the subset, we first selected 10 topics, uniformly at random, from a total of 52 topics with available relevance judgements (out of the 107 topics for 2010). Then, we filtered the relevance judgments, keeping only those regarding the selected topics. Finally, we filtered out documents that were not mentioned in the relevance judgments from each of the four archives (`pages25.tar.bz2`, `pages26.tar.bz2`, `pages27.tar.bz2` and `pages28.tar.bz2`). Our sampling method also provided an option to include all documents linked from the selected documents directly mentioned in the relevance judgments. However, this would result in a much larger dataset than the version without linked documents, defeating the purpose of lowering the scale.



(a) Graph-of-word (document-based graph; text-only). Nodes represent terms. Query term nodes are identified by a thicker border.

(b) Graph-of-entity (collection-based graph; text+knowledge). Pink nodes represent terms and green nodes represent entities. A seed node for the given query is displayed in white. Query term nodes are identified by a thicker border.

Figure 2 Graph-based representations for the first sentence of Wikipedia's "Semantic Search".

From this moment on, any mention to INEX 2009 data refers to the subset that we created and identify as 10T-NL (10 Topics; No Links). This is the dataset that we use in the evaluation process and, while only 10 topics were selected, the subset contains 7,487 documents and 7,504 individual relevance judgments.

4 Representation and retrieval

In our experimental workbench, we implemented the graph-based models using a graph database per index (Neo4j[1]) and the ranking functions using the Gremlin DSL[2]. The goal of this work was to propose a graph-based representation for combined data (text and knowledge), while using the graph-of-word as a text-only baseline. Figure 2 illustrates the graph-of-word and graph-of-entity models, described in the following sections, based on the first sentence of the Wikipedia article for "Semantic Search" (i.e., our example collection consists of only one document with a single sentence):

> *"Semantic search seeks to improve search accuracy by understanding the searcher's intent and the contextual meaning of terms as they appear in the searchable dataspace, whether on the Web or within a closed system, to generate more relevant results."*

4.1 Graph-of-word

The graph-of-word [25] is a document-based graph [7], where each node represents a term and each edge links to the following terms within a window of size n. The graph is unweighted, but directed, defying the term independence assumption of the bag-of-words approach. Figure 2a

[1] `https://neo4j.com/`
[2] Apache Gremlin is a domain-specific language for graph querying. More information at `https://tinkerpop.apache.org/gremlin.html`.

shows a graph-of-word instance for the first sentence of the Wikipedia article on "Semantic Search", using a window size of $n = 3$. The graph-of-word is thus able to capture the context of each term within a particular document.

In the original graph-of-word implementation, the term weight (TW) metric was pre-computed based on the indegree of each term node and stored in the inverted index to be used in place of the term frequency (TF). In our implementation, however, this was done in real time by filtering over the union of all document-based graphs and selecting a given subgraph based on a *doc_id* attribute stored in the edge. This is a less efficient solution, but it simplified the process of exploring and developing the novel graph-of-entity model, based on the graph-of-word, by defining a common representation framework. Additionally, the focus of our experiment was retrieval effectiveness; we were not particularly concerned with index efficiency.

Equation 1 shows the ranking function used for retrieval over the graph-of-word [25, Equation 7].

$$TW\text{-}IDF(t,d) = \frac{tw(t,d)}{1 - b + b \times \frac{|d|}{avdl}} \times log\frac{N+1}{df(t)} \tag{1}$$

The formula was derived from the TF-IDF approach as defined by Lv and Zhai [21], replacing the $tf(t,d)$ function by the $tw(t,d)$ given by the node indegree of term t, for document d, on the graph-of-word. For example, in Figure 2a, we assume the query `[ web search system ]` and find that the largest term weight, $tw(t,d) = 3$, was assigned to "search", while "web" and "system" were tied in second place with $tw(t,d) = 2$. The parameter b was fixed at 0.003, since, according to the authors [25], it consistently produced good results across various collections, with $|d|$ representing the length of document d, $avdl$ the average length of all documents in the corpus, N the number of documents in the corpus, and $df(t)$ the document frequency of term t in the corpus. In our implementation, both $|d|$ and $avdl$ were approximated by the number of edges within the respective document-based graph, since we did all computations directly based on the graph.

4.2 Graph-of-entity

The graph-of-entity is a collection-based graph [7], where nodes can represent either terms or entities and edges can be of three types: term – `[before]`→term, entity – `[related_to]`→entity and term – `[contained_in]`→entity. While the graph-of-entity was inspired by the graph-of-word, it only captures term sequence instead of term context, through term→term relations, that is, the window size is always one. Additionally, we also encode entity→entity relations in the graph as a way of representing knowledge associated with the document (e.g., obtained from an information extraction pipeline applied to the text, or simply consisting of Wikipedia concepts linked in some manner). Finally, term→entity relations are established based on a substring matching approach, where a link between a term and an entity is created whenever the term is contained within the entity's name as a whole word (i.e., partial word matches are not considered). The goal for the first version of this model was to keep it simple (e.g., refraining from using similarity edges), but strongly connected (namely capturing all obvious relations). The main goal was to capture the properties of text, while modeling knowledge and establishing relations between text and knowledge.

We rank entities in the graph-of-entity based on the entity weight (EW) for an entity e and a query q. A set of seed nodes S_q are derived from query q, based on the links between query term nodes and entity nodes; when there are no entity nodes linked to a query term node, then the term node becomes its own seed node. This step provides a representation of

the query in the graph, that will be used as the main input for the ranking function. Next, we present a formal definition for $EW(e, q)$, based on three main score components: coverage $c(e, S_q)$, confidence weight $w(s)$ for a seed node s, and the average weighted inverse length of the path between a seed node s and an entity node e to rank.

Let us assume a graph-of-entity represented by an attributed labeled multigraph G_e, similar to the one depicted in Figure 2b, and a set of operations over G_e to obtain a ranking of entity nodes with a *doc_id* attribute. Let q be a query represented by a sequence of term nodes q_n and let e be an entity node that we want to rank (i.e., it has a *doc_id* attribute). Let Sq be the set of seed nodes derived from query q. For each node q_n that represents a term in query q, we obtain the set of seed entity nodes S_{q_n} that are adjacent to term node q_n. Whenever q_n has no entity node neighbors, $S_{q_n} = \{q_n\}$. The set S_q of all seed nodes derived from query q is then given by $S_q = \bigcup_{q_n} S_{q_n}$. This means that S_q will contain all entity nodes adjacent to query term nodes, as well as query term nodes that are not adjacent to any entity node (i.e., they represent themselves). For example, in Figure 2b, assuming query $q = q_1, q_2, q_3$, the seed nodes are given by $S_q = \{e_1, e_2, e_3, q_3\}$. Let p_{es} be a path between an entity node e and a seed node s, as defined by a sequence of vertices $e, v_1, \cdots, v_{(\epsilon-1)}, s$ in the undirected version of G_e. Let P_{es} be the set of all simple paths p_{es} between e and s. Assume the function $\epsilon(p_{uv})$ as the length of a given path p_{uv} between vertices u and v, representing the number of traversed edges[3].

Equation 2 can be read as the ratio between the number of paths linking entity node e and seed nodes s and the total number of seed nodes S_q. That is, the coverage represents the fraction of reachable seed nodes from a given entity.

$$c(e, S_q) = \frac{|\{s \in S_q | \exists p_{es} \in P_{es}\}|}{|S_q|} \tag{2}$$

Let e_{ts} be the edge incident to both a term node t and a seed node s. Equation 3 can be read as the confidence weight of seed node s. It represents the confidence that a seed node is a good representation of the query term it was derived from.

$$w(s) = \begin{cases} \dfrac{|\{e_{ts} \in E(G_e) | \forall t \exists q (t = q_n)|\}}{|\{e_{ts} \in E(G_e)\}|} & \text{if } s \text{ is an entity node} \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

Finally, Equation 4 shows the ranking function for a given entity e and query q.

$$EW(e, q) = c(e, S_q) \times \frac{1}{|S_q|} \sum_{s \in S_q} \left(\frac{1}{|P_{es}|} \sum_{p_{es} \in P_{es}} w(s) \frac{1}{\epsilon(p_{es})} \right) \tag{4}$$

The query is only used to obtain the seed nodes S_q that best represent q in the graph. This is analogous to a step in a query entity linking process. The remaining steps are quite straightforward. We obtain the average weighted inverse length of the path between each seed node s and the entity e that we want to rank. Assuming that the seed nodes are good representations of the query in the graph, the closer an entity is from all seed nodes, the more relevant it is – closeness is measured by the inverse length of the path. Given there is a degree of uncertainty associated with the selection of seed nodes, we scale this value based

[3] In practice, we also defined a maximum distance threshold to compute the length of a path between two nodes. That is, no paths above the given threshold were considered. For this particular experiment, we used a maximum distance of one, which is an extremely conservative value.

Table 1 Evaluation metrics for the graph-of-word (GoW) and graph-of-entity (GoE) based on INEX 2009 10T-NL (precisions and recall were [macro] averaged over all topics).

Model	P@10	MAP	NDCG@10	Prec.	Recall
GoW	**0.3000**	**0.2333**	**0.3265**	0.1085	**0.9816**
GoE	0.1500	0.0399	0.1480	**0.1771**	0.2233

on the confidence weight of the seed node – an entity close to a high confidence seed node is more relevant than an entity close to a low confidence seed node, but an entity further apart from a high confidence seed node might be on par, or even more relevant.

5 Evaluation

During the evaluation stage, we aimed at assessing the retrieval effectiveness of the graph-of-entity in comparison with a slightly altered implementation of the graph-of-word. Particularly, the document length $|d|$ and the average document length $avdl$, used for pivoted document length normalization [14], were calculated based on the number of term nodes per document, which appear only once per document – this means that we were only able to account for unique terms to obtain the document length in the graph-of-word. However, this change is not particularly critical, given the low sensitivity of the graph-of-word to document length [25, Section 5.3] (using $b = 0.003$ is close to using no pivoted document length normalization at all). That is to say, our implementation of the graph-of-word is only slightly different from the original and still provides a solid baseline.

We prepared two indexes based on the 7,487 documents from the INEX 2009 10T-NL collection, one for the graph-of-word and another one for the graph-of-entity. For our experiment, each index was stored as a graph database. We then retrieved the results for each topic, labeling each entry using a binary relevance attribute based on whether there were any identified passages in the judgments file.

Table 1 shows the result of the assessment for this small subset of INEX 2009 Wikipedia Collection. In particular, we present the precision for the first 10 results (P@10), the mean average precision for a maximum of 1,000 retrieved results (MAP), the normalized discounted cumulative gain for the first 10 results, using binary relevance grades (NDCG@10), and the overall precision and recall. As we can see, the graph-of-word (GoW) obtained the best overall scores, except for precision. Recall for the graph-of-word was nearly optimal (0.9816) and significantly above the recall for the graph-of-entity (0.2233). Such a high recall also translated into a lower precision for the graph-of-word (0.1085), which was the only metric that was beat by the graph-of-entity (0.1771). This means that we were unable to improve graph-of-entity (GoE) over the baseline, as expected. Nevertheless, we obtained a better precision, which is encouraging, given our simplistic first attempt at designing a graph-based representation for combined data.

Given the small dimension of the dataset and in order to better understand the obtained MAP scores, in Table 2 we present the average precision for each topic. We also present the issued query and highlight the highest and lowest scores per model. As we can see, `[ dinosaur ]` achieved the highest average precision in graph-of-word, retrieving 703 results (425 relevant), but only 3 results (all relevant) for the graph-of-entity. The lowest average precision for the graph-of-word was achieved for `[ composer museum ]`, retrieving 1,674 results, out of which only 64 were relevant; this was beat by the graph-of-entity, retrieving 179 results, out of which 30 were relevant. The lowest average precision for the graph-of-entity was achieved for `[ Monuments of India ]`, retrieving only 2 results, none of which were relevant.

Table 2 Average precision per topic for the graph-of-word (GoW) and graph-of-entity (GoE) based on INEX 2009 10T-NL. Highest and lowest average precision per model is shown in bold; results are ordered by decreasing average precision for GoW.

Topic ID	Topic Title (Query)	Average Precision	
		GoW	GoE
2010038	`[ dinosaur ]`	**0.6189**	0.0069
2010057	`[ Einstein Relativity theory ]`	0.2899	**0.1364**
2010003	`[ Monuments of India ]`	0.2888	**0.0000**
2010079	`[ famous chess endgames ]`	0.2541	0.0448
2010023	`[ retirement age ]`	0.2513	0.0027
2010040	`[ President of the United States ]`	0.2408	0.0051
2010096	`[ predictive analysis +logistic +regression model program application ]`	0.2185	0.0410
2010049	`[ European fruit trees ]`	0.0756	0.0119
2010014	`[ composer museum ]`	**0.0624**	0.1185
2010032	`[ japanese ballerina ]`	0.0331	0.0315
	MAP	0.2333	0.0399

While the graph-of-entity clearly captures additional information, differing mainly on the lack of explicit representation of word context, overall it did not present an improvement over the graph-of-word. Our approach focused on assessing the effectiveness of the model, in order to iteratively improve it and eventually surpass existing state-of-the-art graph-based approaches through the integration of text and knowledge and using a collection-based approach. Despite the disregard for efficiency, at this stage, the complexity of the model and its inefficient implementation supported on a graph database were critical challenges in setting up an evaluation workbench with acceptable run times. While we did not index the full INEX 2009 Wikipedia Collection, with over 2.6 million documents, we were able to index a smaller test collection, based on a sample of 10 topics and corresponding judged documents (INEX 2009 10T-NL), in order to obtain some insight. Additionally, during the participation in the TREC 2017 OpenSearch track [11] we had been able to index the complete SSOAR[4] collection and evaluate the models in a real-world scenario, which acts as complementary information to the performance results we present here.

6 Conclusions

We tackled the problem of entity-oriented search through the proposal of a novel graph-based model for the representation and retrieval of combined data (text and knowledge). We proposed a collection-based representation of terms, entities and their relations (term-term, entity-entity and term-entity), as a way to unify unstructured text and structured knowledge as a graph. We then proposed a very basic ranking function, supported on the graph-of-entity, where we mapped the terms of the query into nodes in the graph, preferentially expanding into neighboring entities, in order to obtain a query representation in the graph (seed nodes). We treated this as an open step in an entity linking process, that was only closed during ranking. Ranking was done based on the seed nodes, by treating them as leads. These leads were followed by trying to exhaust all available paths within a maximum distance, which resulted in the scoring of entity nodes. For evaluation purposes, not all entity nodes were

[4] `https://www.gesis.org/ssoar/home/`

ranked, limiting this operation to nodes that directly represented a document in the corpus (e.g., for Wikipedia, the entity mapped to the corresponding article, while, for SSOAR, a special entity had been created to represent the document). This enabled us to map the problem of entity ranking into the domain of documents, thus providing a way to evaluate using the traditional test collections and strategies that were available to us at the time.

The main goal of this work was to provide a simple baseline model that was graph-based and represented combined data in a unified manner. We performed evaluation based on a sample of the INEX 2009 Wikipedia Collection, which complemented the assessments from TREC 2017 OpenSearch track. In particular, we compared the graph-of-entity (our model) with the graph-of-word (a baseline text-only model). Overall, our model could not outperform the baseline, except regarding precision. However, we were able to establish a graph-based strategy to jointly represent combined data, taking into account terms, entities and their relations in order to perform ranking. At the same time, we explored the unification of entity linking and entity ranking as a single task over the graph-of-entity.

References

1 Paavo Arvola, Shlomo Geva, Jaap Kamps, Ralf Schenkel, Andrew Trotman, and Johanna Vainio. Overview of the INEX 2010 Ad Hoc Track. In *Comparative Evaluation of Focused Retrieval - 9th International Workshop of the Inititative for the Evaluation of XML Retrieval, INEX 2010, Vugh, The Netherlands, December 13-15, 2010, Revised Selected Papers*, pages 1–32, 2010. doi:10.1007/978-3-642-23577-1_1.

2 Hannah Bast, Björn Buchhold, Elmar Haussmann, et al. Semantic Search on Text and Knowledge Bases. *Foundations and Trends® in Information Retrieval*, 10(2-3):119–271, 2016.

3 Mikhail Bautin and Steven Skiena. Concordance-Based Entity-Oriented Search. In *The 2007 IEEE / WIC / ACM Conference on Web Intelligence (WI '07)*, pages 2–5, 2007.

4 Michael S. Bernstein, Jaime Teevan, Susan T. Dumais, Daniel J. Liebling, and Eric Horvitz. Direct answers for search queries in the long tail. In *CHI Conference on Human Factors in Computing Systems, CHI '12, Austin, TX, USA - May 05 - 10, 2012*, pages 237–246, 2012. doi:10.1145/2207676.2207710.

5 Ravish Bhagdev, Sam Chapman, Fabio Ciravegna, Vitaveska Lanfranchi, and Daniela Petrelli. Hybrid search: Effectively combining keywords and semantic searches. In *European Semantic Web Conference*, pages 554–568. Springer, 2008.

6 Roi Blanco and Christina Lioma. Random walk term weighting for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 829–830. ACM, 2007.

7 Roi Blanco and Christina Lioma. Graph-based term weighting for information retrieval. *Information Retrieval*, 15(1):54–92, 2012. doi:10.1007/s10791-011-9172-x.

8 Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250, 2008. doi:10.1145/1376616.1376746.

9 Jing Chen, Chenyan Xiong, and Jamie Callan. An Empirical Study of Learning to Rank for Entity Search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 737–740, 2016. doi:10.1145/2911451.2914725.

10 Ruey-cheng Chen, Damiano Spina, W Bruce Croft, Mark Sanderson, and Falk Scholer. Harnessing Semantics for Answer Sentence Retrieval. In *Proceedings of the Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval (ESAIR 2015)*, pages 21–27, 2015.

11 José Devezas, Carla Teixeira Lopes, and Sérgio Nunes. FEUP at TREC 2017 opensearch track: Graph-based models for entity-oriented. In *The Twenty-Sixth Text REtrieval Conference Proceedings (TREC 2017)*, Gaithersburg, MD, USA, 2017.

12 Taoufiq Dkaki, Josiane Mothe, and Quoc Dinh Truong. Passage Retrieval Using Graph Vertices Comparison. In *Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, SITIS 2007, Shanghai, China, December 16-18, 2007*, pages 71–76, 2007. doi:10.1109/SITIS.2007.82.

13 Pedro Domingos. *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Basic Books, 2015. URL: https://books.google.pt/books?id=glUtrgEACAAJ.

14 Hui Fang, Tao Tao, and ChengXiang Zhai. A formal study of information retrieval heuristics. In *SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004*, pages 49–56, 2004. doi:10.1145/1008992.1009004.

15 Miriam Fernández, Iván Cantador, Vanesa López, David Vallet, Pablo Castells, and Enrico Motta. Semantically enhanced information retrieval: An ontology-based approach. *Web semantics: Science, services and agents on the world wide web*, 9(4):434–452, 2011.

16 Shlomo Geva, Jaap Kamps, Miro Lehtonen, Ralf Schenkel, James A. Thom, and Andrew Trotman. Overview of the INEX 2009 Ad Hoc Track. In *Focused Retrieval and Evaluation, 8th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2009, Brisbane, Australia, December 7-9, 2009, Revised and Selected Papers*, pages 4–25, 2009. doi:10.1007/978-3-642-14556-8_4.

17 Udayan Khurana and Amol Deshpande. Efficient snapshot retrieval over historical graph data. In *29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013*, pages 997–1008, 2013. doi:10.1109/ICDE.2013.6544892.

18 Jon M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *J. ACM*, 46(5):604–632, 1999. doi:10.1145/324133.324140.

19 Ching-Pei Lee and Chih-Jen Lin. Large-Scale Linear RankSVM. *Neural Computation*, 26(4):781–817, 2014. doi:10.1162/NECO_a_00571.

20 Bo Lin, Kevin Dela Rosa, Rushin Shah, and Nitin Agarwal. LADS : Rapid Development of a Learning-To-Rank Based Related Entity Finding System using Open Advancement. In *The First International Workshop on Entity-Oriented Search (EOS 2011)*, 2011.

21 Yuanhua Lv and ChengXiang Zhai. Lower-bounding term frequency normalization. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 7–16, 2011. doi:10.1145/2063576.2063584.

22 Bruno Martins and Mário J. Silva. A Graph-Ranking Algorithm for Geo-Referencing Documents. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005), 27-30 November 2005, Houston, Texas, USA*, pages 741–744, 2005. doi:10.1109/ICDM.2005.6.

23 Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

24 Hadas Raviv, David Carmel, and Oren Kurland. A ranking framework for entity oriented search using Markov random fields. In *Proceedings of the 1st Joint International Workshop on Entity-Oriented and Semantic Search (JIWES 2012)*, pages 1–6, 2012. doi:10.1145/2379307.2379308.

25 François Rousseau and Michalis Vazirgiannis. Graph-of-word and TW-IDF: new approach to ad hoc IR. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pages 59–68. ACM, 2013.

26 Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-*

NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003, pages 142–147, 2003. URL: http://aclweb.org/anthology/W/W03/W03-0419.pdf.

27 Ralf Schenkel, Fabian M. Suchanek, and Gjergji Kasneci. YAWN: A semantically annotated wikipedia XML corpus. In *Datenbanksysteme in Business, Technologie und Web (BTW 2007), 12. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme" (DBIS), Proceedings, 7.-9. März 2007, Aachen, Germany*, pages 277–291, 2007. URL: http://subs.emis.de/LNI/Proceedings/Proceedings103/article1404.html.

28 Amit Singhal. Introducing the Knowledge Graph: things, not strings. https://googleblog.blogspot.pt/2012/05/introducing-knowledge-graph-things-not.html, May 2012.

29 Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 697–706, 2007. doi:10.1145/1242572.1242667.

30 Valentin Tablan, Danica Damljanovic, and Kalina Bontcheva. A Natural Language Query Interface to Structured Information. In *The Semantic Web: Research and Applications, 5th European Semantic Web Conference, ESWC 2008, Tenerife, Canary Islands, Spain, June 1-5, 2008, Proceedings*, pages 361–375, 2008. doi:10.1007/978-3-540-68234-9_28.

31 Alberto Tonon, Michele Catasta, Gianluca Demartini, Philippe Cudr, and Karl Aberer. TRank: Ranking Entity Types Using the Web of Data. In *International Symposium on Wearable Computers 2013 (ISWC 2013)*, 2013. URL: http://infoscience.epfl.ch/record/196256/files/TRank.pdf.

32 Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. Word-Entity Duet Representations for Document Ranking. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 763–772, 2017. doi:10.1145/3077136.3080768.

33 Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Ed Hovy. JointSem: Combining query entity linking and entity based document ranking. In *Proceedings of the 26th ACM International Conference on Information and Knowledge Management (CIKM 2017)*, 2017.

34 Nikita Zhiltsov, Alexander Kotov, and Fedor Nikolaev. Fielded sequential dependence model for ad-hoc entity retrieval in the web of data. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253–262. ACM, 2015.